

**FINAL DE ALGORITMOS Y ESTRUCTURAS DE DATOS II**  
**4/3/2021**

**PARA CONSIDERARSE APROBADO DEBEN  
ESTAR BIEN AL MENOS 3 EJERCICIOS**

- 1. (3 puntos)** Se cuenta con la siguiente estructura de datos, que almacena las compras que un conjunto de personas realiza en un supermercado.

supermercado se representa con estr

```
estr = tupla<compras: dicc(persona -> conj(compra)) x  
      productos1: dicc(nat -> string) x  
      productos2: dicc(string -> nat) x  
      ventas_por_prod: dicc(cod_prod -> nat)>
```

donde compra = tupla <cod\_prod: nat x cant: nat>

La primera componente de la tupla compra identifica al código de producto, cuyo nombre aparece buscándolo en el primer diccionario. El segundo diccionario es el inverso del primero. ventas\_por\_prod indica la cantidad total de unidades vendidas para cada producto.

- a)** Escriba en castellano y de manera formal el invariante de representación.  
**b)** Indique cómo representaría ambos diccionarios (productos1 y productos2) en contextos realistas de uso (por ejemplo, sabiendo que la cantidad de productos no está acotada pero que en general tienen nombres breves).
- 2. (2 ptos)** Supongamos una secuencia  $s = \langle s_1, s_2, \dots, s_n \rangle$  de  $n$  enteros positivos que representan el precio de un determinado producto durante  $n$  días consecutivos. Se desea encontrar un par  $(dia\_compra, dia\_venta)$  con  $dia\_compra \leq dia\_venta$ , de tal manera que se maximiza la ganancia obtenida si compramos el producto en  $dia\_compra$  y vendemos en  $dia\_venta$ . Diseñar un algoritmo Divide & Conquer eficiente que resuelva el problema, discutir su complejidad temporal y explicar por qué el algoritmo propuesto es correcto.
- 3. (1 puntos)** ¿Por qué utilizamos la notación  $O$ - $\Omega$ - $\Theta$  para analizar la complejidad de los algoritmos? ¿Para qué se utiliza cada una? Dar ejemplos verdaderos relacionados con algoritmos de Ordenamiento.
- 4. (2 puntos)** Explicar los fenómenos de aglomeración primaria y secundaria. ¿En qué consisten? ¿En qué implementaciones de hash se producen? ¿Cómo se eliminan?
- 5. (2 ptos)** Dada la siguiente especificación (ver página 2), indicar si tiene errores, cuáles y por qué son errores, y cómo los arreglaría (puede axiomatizar o explicar los eventuales arreglos).

## **TAD Empleado**

### **generadores**

crear\_empleado: dni x edad x legajo -> empleado

### **observadores**

= empleado x empleado -> bool

**FIN**

## **TAD Auto es Nat**

## **TAD Empresa**

### **generadores**

crear: conj(empleados) -> empresa

llega\_empleado: empleado x auto x empresa -> empresa

### **observadores**

empleados: empresa -> conj(empleados)

que\_auto\_tiene?: empresa e x empleado p -> auto (p \in empleados(e))

suma\_edades: empresa -> edad

legajo: empresa x empleado -> legajo

### **axiomas**

suma\_edades(crear(c)) = 0

suma\_edades(llega\_empleado(crear\_empleado(d, e, l), a, f)) = e+suma\_edades(f)

que\_auto\_trajo?(crear(c), p) = 0

que\_auto\_trajo?(llega\_empleado(p, a, f), p') = if p=p' then a else que\_auto\_trajo?(f, p') fi

empleados(crear(c)) = c

empleados(llega\_empleado(p, a, f)) = empleados(f)

legajo(e, p)= legajo(p)

**FIN**